

# Práctica 5 - Entrada/Salida

## Organización del Computador 1

Segundo Cuatrimestre 2014

**Ejercicio 1** En una computadora ORGA1 se ha conectado un dispositivo de entrada. Este dispositivo posee un registro de entrada/salida (**STATUS**) cuya dirección se *mapea* a la dirección de memoria 0x0xFFF0.

La forma de interpretar este registro (de 16 bits:  $b_{15} \dots b_0$ ) es la siguiente:

- El bit  $b_{15}$  representa el estado del dispositivo. Dicho bit se encuentra en 1 cuando el dispositivo está ocupado (no hay todavía un dato válido para leer), y en 0 cuando está listo (en caso contrario).
- Cuando el estado del dispositivo es “listo” (es decir, el dato para leer es válido),  $b_7 \dots b_0$  contienen el dato proporcionado por el dispositivo.

Escribir una rutina en assembler de ORGA1 que, implementando la técnica de *polling*, lea datos válidos del dispositivo tantas veces como se indique en R0 y los escriba en forma contigua a partir de la dirección de memoria referida por R1. Por simplicidad, no importa si se lee varias veces el mismo dato o se pierden datos válidos.

**Ejercicio 2** Un procesador ORGA1 se encuentra conectado a 2 dispositivos de E/S:

- Un botón que posee un único registro de E/S de lectura solamente (**BUTTON\_STATUS**). Si se lee el valor 0x0000 el botón no está siendo presionado. Por el contrario, si se lee el valor 0xFFFF el botón está siendo presionado. Este registro de E/S se encuentra mapeado a la dirección 0xFFFF0.
- Un display electrónico que posee un registro de E/S de escritura solamente (**DISPLAY\_DATA**). El valor que se escribe en este registro se muestra en una pantalla al usuario como un número entre 0 y 65535. Este registro está mapeado a la dirección de E/S 0xFFFF1.

Escribir una rutina que controle el procesador tal que, comenzando con el valor 0, cuente la cantidad de veces que el usuario oprime un botón. Debe tener en cuenta que una pulsación del botón incluye esperar hasta que el usuario *suelte* el botón.

**Ejercicio 3** En una computadora ORGA1i conectada a un dispositivo de E/S que la puede interrumpir se está ejecutando el siguiente programa:

```
main:                MOV R0, 0x8000
                    MOV R1, 0xFFFF
                    ADD R0, R1
                    JVS huboOverflow
noHuboOverflow:      ...
                    ...
huboOverflow:        ...
```

La rutina de atención de la interrupción es la siguiente:

```

rutinaAtencionInt:  ...
                   ...
                   PUSH R0          ; salva el valor original de R0
                   PUSH R1          ; salva el valor original de R1
                   MOV R0, 0x0000
                   MOV R1, 0x0000
                   ADD R0, R1
                   POP R1           ; restaura el valor original de R1
                   POP R0           ; restaura el valor original de R0
                   SETI             ; habilita que se pueda interrumpir
                   RET              ; retorna a lo que estaba ejecutando

```

Suponga que  $I=1$  y en la dirección de memoria 0x0000 se encuentra efectivamente la dirección de la etiqueta `rutinaAtencionInt`

- ¿La ejecución del programa `main` continuará como se espera si el dispositivo de E/S activa la señal de interrupción durante el ciclo de ejecución de la instrucción `ADD R0, R1`?
- ¿Qué cambios realizaría a la rutina de atención de la interrupción para que su ejecución fuera transparente?

**Ejercicio 4** Una computadora ORGA1 se encuentra conectada a 3 dispositivos de E/S que actúan como sensores. Cada sensor posee un registro de E/S que reporta información climática. Los sensores son:

- Temperatura: Mide temperatura en grados celsius (TEMP\_STATUS)
- Presión Atmosférica: Mide presión en HectoPascuales (ATM\_STATUS)
- Intensidad del Viento: Mide intensidad en Metros/Segundo (WIND\_SPEED)

Las etiquetas `MAX_TEMP`, `MAX_ATM`, `MAX_WIND_SPEED` son constantes de 16 bits.

- Mapear los registros de E/S a direcciones de E/S de ORGA1 y completar el siguiente código:

```

senzar: CMP [...], MAX_TEMP ;alcanzo temperatura maxima?
        JG alarma
        CMP [...], MAX_ATM ;alcanzo presion atmosferica maxima?
        JG alarma
        CMP [...], MAX_WIND_SPEED ;alcanzo velocidad maxima?
        JG alarma
        JMP senzar
alarma: CALL sonarAlarma ;invoca rutina de alarma
        JMP senzar

```

- Suponiendo que el ciclo de instrucción de cada instrucción del programa tarda  $t$  sg y los valores máximos nunca se alcanzan ¿cuál es la frecuencia (en Hz) de muestreo (lectura) de los registros de E/S?
- Suponga que reemplaza el procesador ORGA1 por un procesador ORGA1i que soporta hasta una interrupción. El sensor de temperatura solicita una interrupción si se alcanza la temperatura máxima.
  - Modificar el programa presentado para aprovechar esta característica de modo que la frecuencia de muestreo sea mayor. ¿Cuál es la nueva frecuencia de muestreo en Hz?
  - Escribir la rutina de atención de la interrupción del sensor de temperatura.

**Ejercicio 5** El siguiente pseudo-código incompleto describe el ciclo de instrucción de un procesador ORGA1i :

```
while (true) {
    Fetch()    // fetch
    Decode()   // decodificacion
    Execute()  // ejecucion
    if I==1 AND INTR==1 {
        ... Completar
        INTA()
    }
}
```

Completar el pseudo-código de modo que describa correctamente las acciones que lleva a cabo el procesador cuando decide atender una interrupción.

**Ejercicio 6** Se posee un procesador ORGA1i para implementar un sistema de control de la barrera de un estacionamiento. El funcionamiento esperado del sistema es el siguiente:

- 1) El automovilista presiona un botón que activa el motor de apertura de la barrera.
- 2) Una vez que el vehículo cruza la barrera un sensor especialmente ubicado activa el motor para cerrar la barrera.
- 3) La barrera sólo funciona 12 horas al día (desde el momento de encendido del sistema). Si el automovilista presiona el botón fuera del horario de atención, la barrera no se levanta.

Además del microprocesador ORGA1, se cuenta con los siguientes dispositivos de E/S:

- Un **botón** con un registro de E/S (BUTTON\_DATA). Si se lee el valor 1 del registro de E/S significa que el botón está siendo presionado por el automovilista. Si por el contrario, se lee el valor 0, significa que el botón no está siendo presionado.
  - Un **motor** con un registro de E/S (MOTOR\_CTRL) para controlar la barrera. Si se escribe el valor 1 en su registro de E/S, el motor levanta la barrera. Si por el contrario, se escribe el valor 0, el motor baja la barrera.
  - Un **sensor** con un registro de E/S (SENSOR\_DATA). Si se lee el valor 1 del registro de E/S quiere decir que el sensor está captando un automóvil. En cambio, si se lee el valor 0, no hay ningún automóvil en el rango del sensor.
  - Un **clock** que emite un pulso por una línea de salida (llamada CLOCK\_OUT) cada 10 minutos.
- a) Graficar un esquema de conexión del sistema indicando claramente como conectaría los registros de los dispositivos a los puertos de E/S de la máquina ORGA1i . No olvide graficar la línea de interrupción.
  - b) Escribir la rutina de atención de interrupciones para controlar el paso de las 12 horas. (**Ayuda:** 12 horas equivalen a 720 minutos).
  - c) Escribir (primero en pseudo-código y luego en lenguaje ensamblador) la rutina de control del sistema. Recordar que la barrera sólo debe funcionar dentro del horario establecido.
  - d) Suponiendo que el microprocesador puede ejecutar 750 millones de instrucciones por segundo. ¿Cuánto tardaría en ejecutarse la rutina de atención de interrupciones ?

**Ejercicio 7** El instrumental de navegación de un avión posee los siguientes monitores:

Prioridad	Monitor	Dirección RAI
1	Presión del aire en cabina	0xFF00
2	Altura	0xA000
3	Combustible	0xFE00
4	GPS	0x01FF

Cada monitor levanta una señal cuando registra un cambio en el entorno que está midiendo. Un monitor de mayor prioridad debe poder interrumpir la rutina de atención de interrupción de un monitor de menor prioridad.

Suponiendo que la cabina utiliza un procesador INTEL 8086 con un PIC 8259A:

- ¿En qué orden conectaría las entradas del PIC a las salidas de los monitores?
- ¿Cuál debería ser el valor inicial del registro IMR?
- Describe los valores almacenados en el vector de interrupciones.
- Si el monitor de Altura y de GPS levantan la señal de actualización de datos, ¿qué valor tendría el registro IRR?
- Completar la siguiente rutina de atención:

RAT\_MONITOR\_ALTURA:

```

; Guarda la mascara
    PUSH AX                ; apila el registro AX
    IN AX, IMR             ; copia el contenido del reg. de ES IMR en AX
    PUSH AX                ; apila la mascara actual

; Setear mascara inhibiendo interrupciones de menor prioridad.
    MOV AX, .... (completar)
    OUT IMR, AX            ; copia el contenido de AX en el reg. de ES INTMASK

; habilitar interrupciones
    ....(completar)

; Salvar el estado de lo que falte
    ....(completar)

; Obtener la altura nueva
    IN AX, 43h             ; copia el contenido del reg. 43h de ES en AX
    LEA SI, MONITOR_ALTURA ; Copia la constante MONITOR_ALTURA en el reg. SI
    MOV [SI], AX           ; Copia el contenido del registro AX en la
                           ; direccion apuntada por el reg. SI

; Ya se actualizo la altura, completar lo que falta para terminar.

```

- ¿Es necesaria una instrucción especial para indicar el fin de la ejecución de una rutina de atención de interrupción o alcanza con el mismo RET que se utilizar para volver de un CALL?

**Ejercicio 8** Un procesador 8086 tiene conectados tres dispositivos de E/S:

INTR	Dispositivo E/S	Tiempo ejecución RAI
2	impresora	10 <i>t</i>
4	disco rígido	8 <i>t</i>
5	puerto serie RS232	7 <i>t</i>

Para administrar las interrupciones se conecta el procesador a un PIC 8259. Las prioridades de estos dispositivos son ascendentes y el valor inicial del registro IMR es 0x00FF. La programación de cada rutina de atención de interrupción asegura que cuando se está atendiendo un dispositivo de prioridad  $n$ , las interrupciones de menor prioridad son enmascaradas (ignoradas). Por otro lado, si se está atendiendo un dispositivo de prioridad  $n$  e interrumpe un dispositivo de prioridad mayor, la interrupción de mayor prioridad se atiende.

Dada la siguiente secuencia de eventos:

- 1)  $t = 10$ : la impresora envía un pedido de interrupción.
- 2)  $t = 15$ : la línea RS232 envía un pedido de interrupción.
- 3)  $t = 20$ : el disco envía un pedido de interrupción.
- 4)  $t = 40$ : el disco envía un pedido de interrupción.

Completar la tabla mostrando a qué se dedica la CPU y cuál es el valor del registro IMR. Suponer que cuando la CPU no está atendiendo interrupciones se encuentra corriendo un programa de usuario.

t	solicita interrupción	valor IMR	actividad CPU
...	...	...	...

**Ejercicio 9** Se desea implementar el controlador de interrupciones PICorga. Este controlador de interrupciones ofrecerá las funcionalidades de un PIC para un procesador ORGA1i. Como hay sobreabundancia de procesadores ORGA1, los ingenieros han decidido construirlo utilizando un procesador ORGA1.

Para ello se tomaron las siguientes decisiones de diseño para modelar el comportamiento de un PIC usando un procesador ORGA1:

dirección	modela	comportamiento
0xFFFF0	IMR	máscara de interrupciones
0xFFFF1	IRR	dispositivos solicitando interrupción
0xFFFF2	bus de datos	el PIC indica la interrupción a atender
0xFFFF3	INTR	escribir 1 en el bit menos significativo para solicitar interrupción al procesador
0xFFFF4	INTA	el procesador escribe 1 en el bit menos significativo para indicar que atiende la interrupción

Escribir el pseudo-código y el programa en lenguaje ensamblador ORGA1 que ejecuta el PICorga.

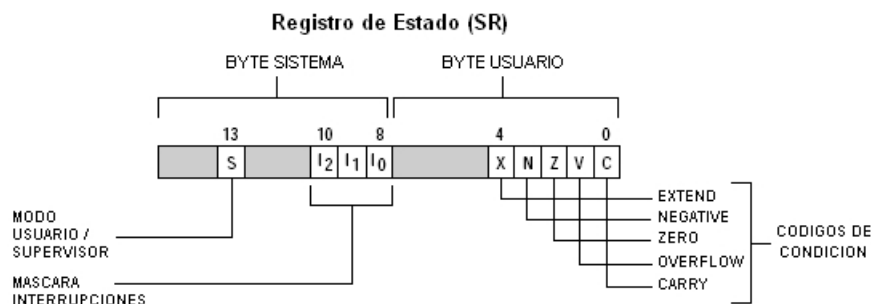
**Ejercicio 10** El MC68000 es un procesador que contiene un PC de 24 bits que direcciona a byte y un registro de estado (SR) de 16 bits el cual guarda el estado del procesador y los flags.

Este procesador posee 2 modos de operación que son indicados en el bit S del registro SR:

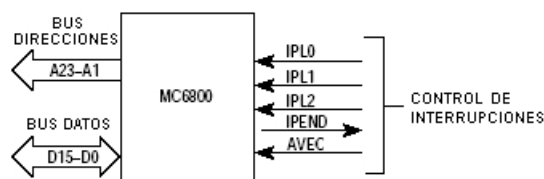
- Usuario ( $S=0$ ): sólo se puede acceder a los primeros 8 bits del SR que contienen los flags.
- Supervisor ( $S=1$ ): se puede acceder a todo el SR (incluyendo a la máscara de interrupciones que contiene la prioridad y el bit de modo) y a direcciones de memoria donde están mapeados los dispositivos de E/S.

Los bits I2,I1,I0 del registro de estado definen la máscara de interrupciones utilizada.

Máscara	Significado
111	máxima prioridad
...	...
001	mínimo prioridad
000	interrupciones deshabilitadas



**Esquema Simplificado MC68000**



El CPU posee:

- 3 señales de entrada: IPL0, IPL1 e IPL2 para las interrupciones externas. Estas señales indican el nivel de prioridad de la interrupción solicitada. El CPU atiende una interrupción sólo si  $I_2, I_1, I_0 \neq 000$  y  $I_2, I_1, I_0 \leq IPL_2, IPL_1, IPL_0$ .
- 1 señal de salida: IPEND, para el reconocimiento de la interrupción. Si el CPU reconoce la interrupción levanta la señal (IPEND=1).

Una vez que el procesador reconoció la interrupción (IPEND=1), utiliza la dirección del vector de interrupciones 000000h, e indexa esta tabla usando el nivel de interrupción codificada en IPL2, IPL1, IPL0

- a) Detalle todos los pasos que cree usted que se desencadenan en el sistema al llegar un pedido de interrupción hasta que la misma es satisfecha y se devuelve el control. Justifique la decisión del orden de los pasos y aclarando debidamente que es lo que realiza el hardware y que es lo que realiza la rutina de atención de interrupciones (software).
- b) Con los pasos que detalló en 1 describa el comportamiento del 68000 si se activan las líneas IPLx de la siguiente forma  $IPL_2=1, IPL_1=0, IPL_0=0$ , con un  $SR=021Ah$ . Mencione a qué posición de memoria se va a buscar la dirección de inicio de la rutina de atención de interrupciones de este dispositivo.
- c) Realice un gráfico que muestre que interrupción se esta atendiendo, en que estado queda el SR a cada instante suponiendo que su valor inicial es  $SR=0111h$ . Utilice el número de dispositivo como su nivel de interrupción. Suponga que las rutinas no alteran los códigos de condición y aquellos bits que no se utilizan permanecen en 0. Tome en cuenta que el microprocesador puede encolar hasta 2 pedidos de interrupción y que atender la interrupción le lleva 10 unidades de tiempo. Mientras que no esta atendiendo una interrupción el 68000 corre un programa con nivel de usuario.

Instante T	Dispositivo
10	1
25	4
30	6
43	2
44	3

**Ejercicio 11** Se desea diseñar un nuevo modelo del procesador ORGA1i que soporte 2 niveles de interrupciones enmascarables: uno de prioridad alta (HP) y otro de prioridad baja (LP).

El flag I se reemplaza por 2 nuevos flags: HIF y LIF que habilitan y deshabilitan las interrupciones de prioridad alta y baja respectivamente. A su vez, los flags IFH y IFL indican si se solicitó una interrupción de prioridad alta o de prioridad baja respectivamente.

La dirección de inicio de la rutina de atención de la interrupción HP se encuentra en la dirección de memoria 0x0000 y la dirección de inicio de la rutina de atención de la interrupción LP se encuentra en la dirección 0x0001.

- Describir en pseudocódigo los pasos que realiza atómicamente el hardware del procesador desde que termina de ejecutar la instrucción actual hasta dejar paso a la rutina de atención de interrupción correspondiente.
- Explique qué instrucciones debe agregar o modificar para completar la arquitectura del nuevo procesador.

**Ejercicio 12** Un nuevo prototipo de horno microondas usará el procesador ORGA1i con soporte para 2 niveles de interrupción diseñado en el ejercicio anterior.

- En la línea LP se conecta un Reloj de Tiempo Real (RTC) y un Panel de Control (CP). Por lo tanto cualquiera de los dos puede generar una interrupción de prioridad baja. Las rutinas que manejan estos dispositivos se encuentran en las direcciones 0x0108 y 0x0304h respectivamente.
- En la línea HP se conecta un sensor de apertura de puerta (DS). La rutina que maneja este dispositivo se encuentra almacenada a partir de la dirección 0x0510.

Los dispositivos RTC y el CP poseen sendos registros de estado (RTC.STATUS y CP.STATUS). Ambos dispositivos continúan solicitando interrupción hasta que se escribe la constante 0x0000 en su registro de estado o el dispositivo desiste del pedido.

- Describa en pseudo-código y en lenguaje ensamblador la rutina de atención de interrupciones de prioridad baja (mostrar cómo detecta quién produjo la interrupción).
- Cuando se termina la rutina de atención de interrupción, ¿qué operaciones necesita realizar el procesador para retornar de la interrupción en forma transparente?

**Ejercicio 13** Ordenar por prioridad decreciente los pedidos de interrupción de los siguientes dispositivos de E/S. Justificar brevemente.

- Reloj del sistema
- Disco Rígido
- Teclado
- Disco Flexible
- Puerto serial
- Impresora

**Ejercicio 14** En una computadora ORGA1 se ha conectado un controlador DMA. El acceso a cada uno de los registros del controlador está mapeado a memoria del siguiente modo:

DEVICE_ID	DEVICE_ADDRESS	MEM_ADDRESS	SIZE	STATUS
0xFFFF0	0xFFFF1	0xFFFF2	0xFFFF3	0xFFFF4

Para efectuar la transferencia hay que indicar el identificador de dispositivo, la dirección dentro del espacio de direcciones del dispositivo identificado, la dirección de memoria y la cantidad de palabras a transferir.

El identificador de la cinta es 0x354A. Se quiere transferir desde la posición 0x0045 de la cinta hasta la 0x013A. Estos datos se deben guardar a partir de la posición 0xA142 de la memoria principal. El bit menos significativo del registro **STATUS** contiene un 1 en caso de escritura, y un 0 en caso contrario. La forma de indicarle al controlador DMA que ya se cargaron todos los datos necesarios, y que puede iniciar la transmisión, es seteando en 1 el bit más significativo del registro **STATUS**. Escribir un programa en *assembler* que acceda a una cinta para transferir datos via DMA.



## Arquitecturas de Referencia

### Descripción General

#### Procesador ORGA1i

El procesador ORGA1i es un procesador ORGA1 que ha sido extendido con la capacidad para atender la interrupción de un único dispositivo de E/S. Para ello:

- Posee una señal de entrada llamada INT (Interrupción) y de salida INTA (Interrupción Reconocida).
- Posee un nuevo flag: I. Este flag indica:
  - si  $I=1$  el procesador puede ser interrumpido y atender la interrupción del dispositivo
  - si  $I=0$  el procesador no puede ser interrumpido
- Todos los flags (I,C,V,N,Z) se almacenan en un registro de uso específico PSW.
- La dirección de memoria 0x0000 se reserva para almacenar la dirección de la rutina de atención de la interrupción del dispositivo de E/S.

Las señales INTR e INTA conectan al procesador con el dispositivo de E/S. Si el dispositivo activa la señal de interrupción y el flag I vale 1, al terminar de ejecutar la instrucción en curso, el procesador realiza *átomicamente* la siguiente secuencia de pasos:

1. Coloca  $[SP]=PSW$ , y decrementa el SP ( $SP=SP-1$ ).
2. Coloca  $[SP]=PC$ , y decrementa el SP ( $SP=SP-1$ ).
3. Coloca  $I=0$  para evitar que el procesador vuelva a interrumpirse.
4. Coloca  $PC=[0x0000]$ .
5. Activa la señal INTA para indicarle al dispositivo que atenderá su pedido.

Luego de esto, comienza a ejecutarse la rutina de atención de la interrupción propiamente dicha.

El conjunto de instrucciones del procesador incluye las instrucciones:

- CLI que coloca el flag  $I=0$
- STI que coloca el flag  $I=1$
- PUSH  $R_i$ , cuyo efecto es  $[SP]=R_i$  y luego  $SP=SP-1$
- POP  $R_i$ , cuyo efecto es  $SP=SP+1$  y luego  $R_i=[SP]$
- IRET, cuyo efecto es  $PC=[SP], PSW=[SP+1], SP=SP-2$

#### Procesador 8086 + PIC 8259<sup>1</sup>

El procesador 8086 posee una arquitectura de 16 bits con direccionamiento a byte. Sus características más sobresalientes (para la realización de esta guía) son las siguientes:

**Registros** cuenta con 12 registros de 16 bits, 8 de propósito general (llamados: AX, BX, CX, DX, BP, SP, DI y SI) y cuatro para manejo de memoria (CS, DS, SS, ES). Puede accederse a la parte baja y alta de los cuatro primeros utilizando AL, BL, CL y DL, y AH, BH, CH y DH.

---

<sup>1</sup>**Aclaración:** en esta sección se desarrolla una descripción incompleta de las características de la arquitectura Intel 8086, pero que es suficiente para realizar los ejercicios de esta guía de ejercicios.

**Flags** provee los que existen en la ORGA1i y algunos más (que no vienen al caso en este momento).

**Instrucciones** las operaciones de la ORGA1i son un subconjunto de las del Intel 8086, pero deben tenerse en cuenta las siguientes particularidades:

- el único modo de direccionamiento para acceder a memoria es el directo;
- a lo sumo uno de los operandos puede ser una dirección de memoria;
- a lo sumo uno de los operandos puede ser una constante;
- para acceder a los registros de E/S se utiliza un espacio de direcciones independiente al de memoria (al contrario de lo que ocurre con la ORGA1i), para lo cual la arquitectura 8086 provee las instrucciones especiales:
  - **IN Reg,RegES** que copia el contenido del registro de E/S a un registro del CPU
  - **OUT RegES,Reg** que copia el contenido del registro en el registro de E/S.

El PIC 8259 es un módulo cuyo objetivo es gestionar distintas interrupciones para el procesador 8086. Cuenta con ocho entradas IR0 a IR7, donde IR0 es la interrupción de mayor prioridad e IR7 es la interrupción de menor prioridad. El PIC contiene los siguientes registros de E/S:

- **IRR** (Interrupt Request Register): El i-ésimo bit se activa si la i-ésima línea de interrupción es activada.
- **IMR** (Interrupt Mask Register): Permite indicar qué interrupciones deben ser atendidas y que interrupciones no deben serlo. El i-ésimo bit prendido indica que la i-ésima interrupción debe ser atendida. Si el bit está apagado, la interrupción se desestima.

Cuando el PIC solicita una interrupción al CPU:

1. Si el CPU decide atender la interrupción, levanta la señal INTA (interrupción reconocida).
2. Al detectarlo, el PIC coloca en el bus de datos el número de interrupción que corresponde atender.
3. El CPU lee el número de interrupción del bus de datos, e indexa el vector de interrupciones (tabla que comienza en la dirección 0x0000) para obtener la dirección de inicio de la rutina de atención para la interrupción solicitada.
4. El CPU apila la palabra de estado, el program counter y deshabilita todas las interrupciones.
5. El CPU inicia la ejecución de la rutina de atención de la interrupción.